OF INNOVATION: TOWARD A NEW BOND BETWEEN STANDARDS DEVELOPMENT ORGANIZATIONS, INDUSTRY FORA, AND OPEN-SOURCE SOFTWARE PROJECTS

The authors examine the evolving roles of open-source software communities, industry fora, and standards development organizations, and places them in an NFV/SDN context. They sketch the differences between these roles and provide guidelines on how the interaction between them can turn into a mutually beneficial relationship that balances the conflicting goals of timely development on the one hand and technical excellence, openness, and fairness on the other, to reach their common goal of creating flexible and efficient telecommunications networks.

Bram Naudts, Wouter Tavernier, Sofie Verbrugge, Didier Colle, and Mario Pickavet

ABSTRACT

Standards development organizations (SDOs) exist to assure the development of consensus-based, quality standards. These formal standards are needed in the telecommunications market to achieve functional interoperability. The standardization process takes years, and then a vendor still needs to implement the resulting standard in a product. This prevents service providers (SPs) who are willing to venture into new domains from doing so at a fast pace. With the development of software-defined networking (SDN) and network function virtualization (NFV), open-source technology is emerging as a new option in the telecommunications market. In contrast to SDOs, open-source software (OSS) communities create a product that may implicitly define a de-facto standard based on market consensus. Therefore, SPs are drawn to OSS, but they face technical, procedural, legal, and cultural challenges due to their lack of experience with open software development. The question therefore arises, how the interaction between OSS communities, SDOs, and industry fora (IF) can be organized to tackle these challenges.

This article examines the evolving roles of OSS communities, IF, and SDOs, and places them in an NFV/SDN context. It sketches the differences between these roles and provides guidelines on how the interaction between them can turn into a mutually beneficial relationship that balances the conflicting goals of timely development on the one hand and technical excellence, openness, and fairness on the other, to reach their common goal of creating flexible and efficient telecommunications networks.

The authors are with Ghent University – iMinds.

Based on the number of subscribers and the multibillion dollar industry that surrounds it, we can resolutely state that fixed and mobile network architectures are very successful. These architectures are fit-for-purpose closed systems based on standardized interfaces. Every component performs specific functions, and each of the dozens of interfaces has a unique definition that has been standardized via an often long, formal, and consensus-based procedure. However, as customer demand evolves and new technologies emerge, the complex nature of these architectures starts to become a hindrance to sustainable growth. First, SPs will have to deal with higher capital expenditures and operational expenditures at a time when average revenue per user is decreasing [1]. As a result, some SPs will delay or refrain from investing further while those who do invest in new services or features face long time-to-market periods as they push an entire industry to standardize the newly developed features and then wait for vendors to actually implement them [1]. Furthermore, even when these new features are standardized and implemented, it may not be possible to realize them with existing equipment, as even though these can be controlled through standardized interfaces, there is little possibility to extend them through the use of open interfaces such as extensible application programming interfaces (APIs).

Therefore, SPs are looking for alternatives that can reduce the time-to-market and cost of new products and services. Three complementary, self-reinforcing drivers can bring them closer to that goal. First, the shift toward SDN offers the opportunity to learn from the experience of previous and ongoing management domain endeavors so as to be able to move to the next level of insight in realizing truly open and extensible interfaces. Additionally, there is an opportunity to migrate from multiple operations systems silos and many specialized operations functions in SP networks toward operations support systems that provide an overall solution architecture for operating services delivered across current and new technologies. Second, NFV can decrease the dependence on expensive network equipment vendor solutions, by replacing network functions with software implementations running on low-cost multi-purpose hardware. The advantages of NFV are most relevant for location independent network functions as better service scalability can be realized through sharing of resources. Third, by investing in OSS, a de-facto market-based standard can be created while the software is developed, and the time-to-market can be reduced by providing a workflow that allows for rapid deployment of software updates to very flexible hardware platforms. However, OSS development also faces challenges such as poor interoperability and high integration costs.

These de-facto market-based standards compete with the telecommunications market's long and often successful tradition of consensus-based standards that are developed within SDOs and

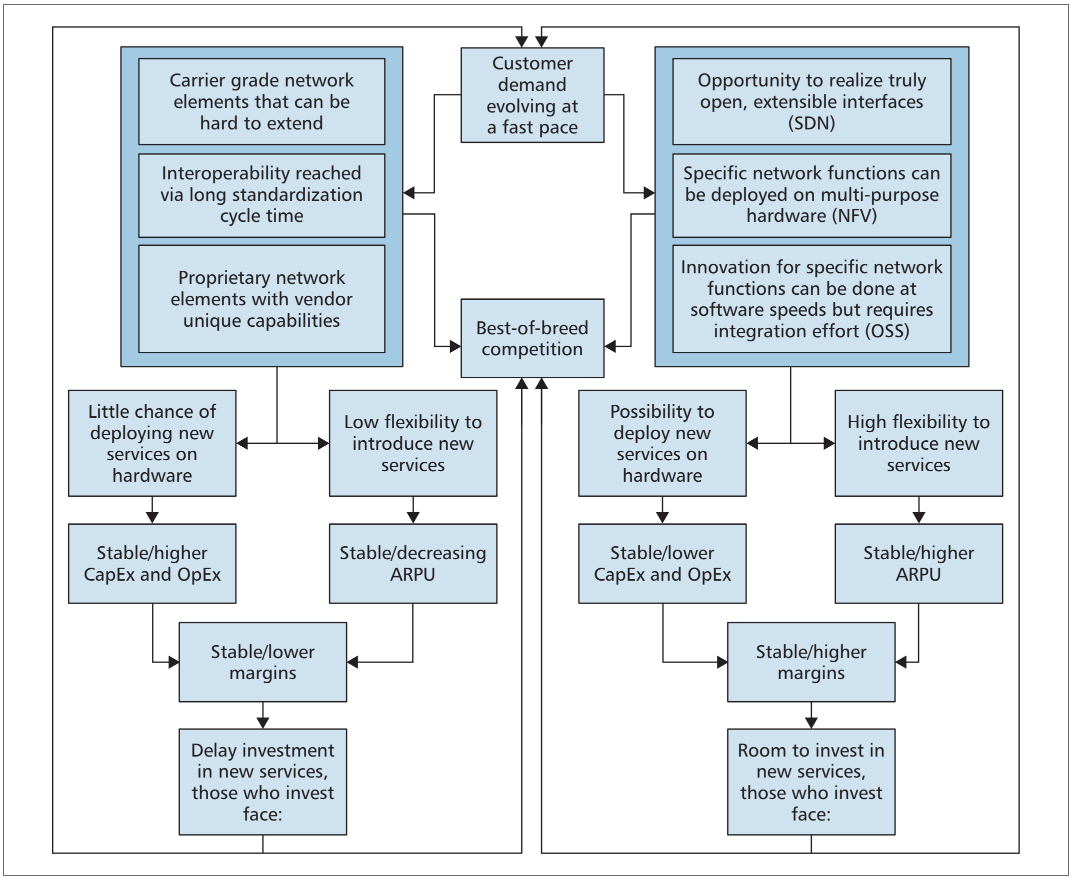

Figure 1. The operator's perspective: benefits and drawbacks of continuing with conventional methods versus the benefits and drawbacks of migrating to SDN/NFV and OSS.

> The general trend toward open source, particularly open APIs, and the interest of SPs in these can be seen as a reaction to the lack of attention to the operational reality that SPs face day in and day out and the domination of vendors and academics in the decision-making processes within the SDOs.

IF. The general trend toward open source (OS), particularly open APIs, and the interest of SPs in these can be seen as a reaction to the lack of attention to the operational reality that SPs face day in and day out and the domination of vendors and academics in the decision-making processes within the SDOs [2]. Even though the strength of the carrier voice varies across SDOs/IF, some SDOs recognize this challenge. The Internet Engineering Task Force (IETF), for example, has a network working group that addresses the perceived gap between operators and the IETF whose objective is to help ensure that operational realities inform the development of key standards [3]. According to a survey conducted by that working group among network operators, the culture within the SDO was given as one of the four major obstacles to participation (time, money, and awareness are the other three) [3]. While the IETF is open to participation by anyone, almost half of the respondents avoid that organization because they do not feel their operator input is welcomed [3]. By not engaging, network operators write themselves out of the process, leading to the disparity that operators are expected to deploy technologies of which they do not even know that the standards are being developed. A recent counter example to the lack of involvement of SPs is the standardization process of NFV at the European Telecommunications Standards Institute (ETSI) which was initiated by Internet Service Providers (ISPs).

Without a doubt, SDOs are needed to produce high quality, relevant technical and engineering documents that create flexible and efficient telecommunications networks. However, standards become less relevant if they trail behind the pace of technology evolution. As such, if the trend toward OSS projects continues, the question arises how SDOs/IF can remain relevant in their role of enabling innovation. The goal of this article is to describe how the interaction between OSS communities, SDOs, and IF can be improved. The remainder of this article is structured as follows. After introducing an overarching SDN/NFV architecture and describing the most relevant roles in the ecosystem, we discuss the differences between the market-based standards formed in OSS communities and the consensus-based standards developed by SDOs/IF. We then formulate guidelines on how these can work together to reach a mutually beneficial relationship.

SDN/NFV Architecture Overview and Main Ecosystem Roles

This section sketches the main functional components and layers in the control architecture of a modern telecom network supporting NFV and links them to the main ecosystem roles, in order to provide the necessary context for the discussion on the interaction between OSS communities, SDOs, and IF. The International Telecommunications Union Telecommunication Standardization Sector (ITU-T) describes

Modern network architectures are structured into multiple functional layers of smaller components. This modular approach reduces complexity, enhances component reusability, and enables multiple migration paths toward future architectures.

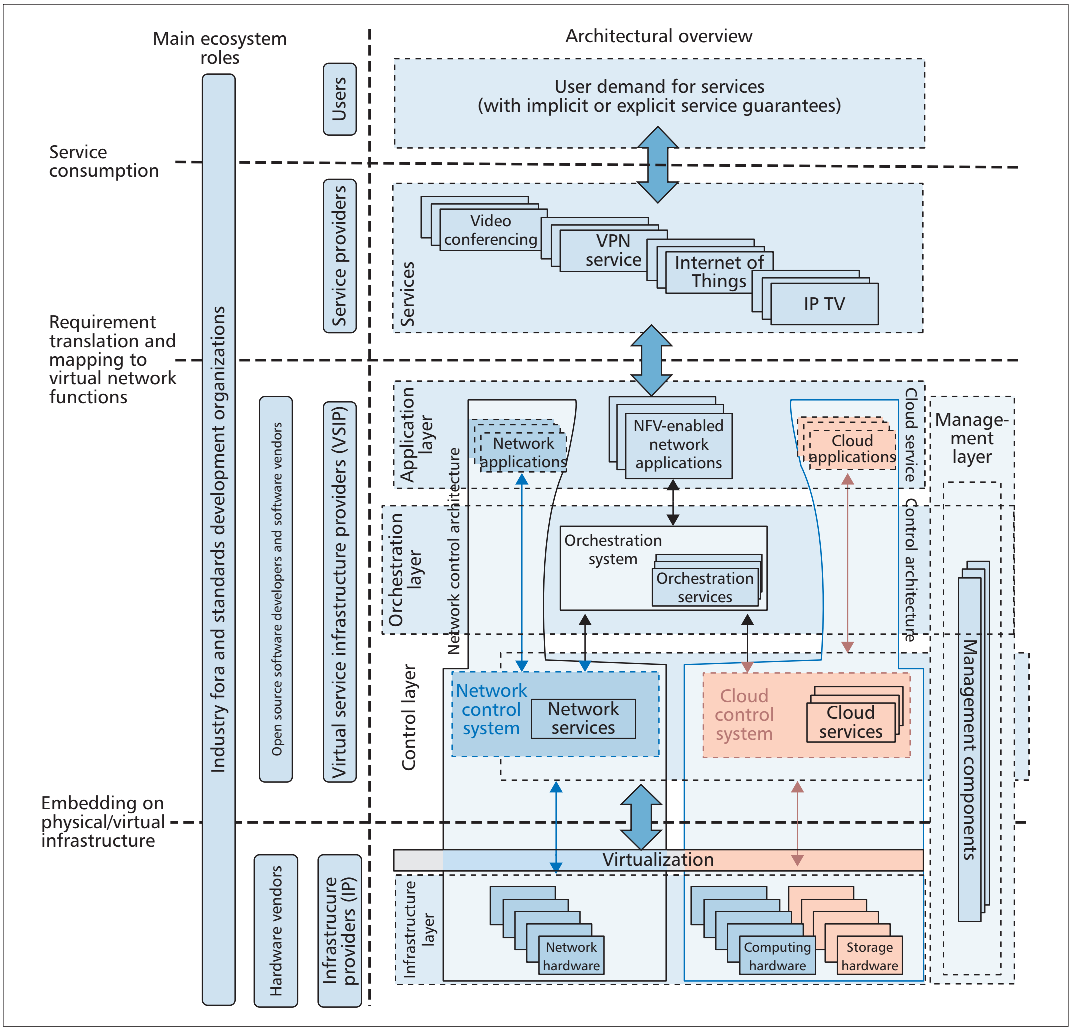

Figure 2. Architectural overview of network- and cloud control platforms.

the requirements to reach carrier grade service for an independent, scalable control plane in future, packet-based networks [4]. The requirements include reachability, scalability, flexibility, reliability, manageability, service, security, interworking, routing, and forwarding.

Modern network architectures are structured into multiple functional layers of smaller components. This modular approach reduces complexity, enhances component reusability, and enables multiple migration paths toward future architectures. Recent softwarization and virtualization tendencies have only further accumulated the decomposition of functional components and layers within architectures. By decoupling the forwarding from control functionality, SDN transforms previously monolithic switches/routers into multiple independent components. Server and network virtualization mechanisms in turn introduce additional functional splits that isolate the data plane functionality of its underlying hardware platform (interested readers should consider [5] and references therein). When network functions (NFs) such as firewalls (FW) or deep packet inspectors (DPI) are decoupled from their underlying hardware platform, and are realized in software that might be executed by commercial off-the-shelf (COTS) hardware, we are speaking about NFV.

SDN and NFV are fully complementary paradigms [6]. SDN is centered on the software-based control of network resources to provide services, while NFV focuses on the creation and life cycle support of some classes of service resources, i.e. virtualized NFs. Indeed, a software-based control architecture might be used to provide network services that consist of either traditional network hardware, virtualized network resources, or combinations of both. In fact, such a combination might be conceived by considering two existing control areas:

- The (software-driven[1]) control of communication networks.
- The control of cloud (service) platforms.

Both control architectures are depicted in the architectural overview of Fig. 2, which is based on [7].

The first (in blue, left) is in charge of controlling the network of switching and routing equipment; the second (in orange, right) is in charge of creating and exposing cloud networks, i.e. a network of reusable computing and storage servers for the purpose of building web services, for example. The control architecture of both domains follows a roughly similar three-layered approach, as depicted in Fig. 2. At the lowest layer, infrastructure resources form the physical foundation on top of which services are provided.

[1] In the context of this article we focus on SDN-controlled networks, although traditional distributed routing protocols could also be considered as the control layer of communication networks.

Communication networks rely on network hardware such as switches and routers; cloud infrastructures rely on (interconnected) computing and storage hardware (servers). A second layer, the control layer, interconnects the components of the infrastructure layer via their north-bound interface (e.g. OpenFlow for network control) in order to provide control-level services such as topology management or datastore services.

The virtualization layer enables a decoupling of functionality from its underlying hardware. At the computing device level, virtualization enables one device to be segmented in multiple logical devices. At the network level network virtualization enables isolation of network resources across different network hardware devices into virtual networks or slices.

At the highest layer, components of the application layer build further on control layer services to program client applications. A traffic engineering application might be defined on top of the SDN-control layer, while a Hadoop cluster might be an application on top of the cloud platform. The orchestration system has a complete view on available networking as well as on computing and storage resources, and is used for services that require a combination of these resources. The orchestration components are able to make an informed decision on which infrastructure should be used. The provisioning process itself can then be further delegated to the already existing network and cloud control system. Orthogonal to the horizontal layers, management functionality might be required to configure any of the components at the infrastructure, control, or application layer, for example to ensure policies or security-related options.

A number of stakeholders are involved in the realization of this SDN/NFV-driven architecture. We discuss stakeholder responsibilities and interactions in the remainder of this section. On the left side of Fig. 2, the most relevant ecosystem roles are represented. These roles are accomplished by the actors that actively participate in the exchange of value. Most actors will perform more than one role at the same time. For example, traditional ISPs fulfill the role of infrastructure provider, virtual service infrastructure provider, and service provider.

Users: Users, i.e. end/enterprise users, retail, or over-the-top providers, request, and consume a diverse range of services. In general, users have no strong opinion about how the service is delivered as long as their quality of experience expectations are satisfied.

Service Providers (SPs): SPs accommodate the service demand from users by offering one or multiple services, including over-the-top service and X-play services (e.g. triple play). The service provider realizes the offered services on a (virtualized) infrastructure via the deployment of virtualized network functions (VNFs).

Virtual Service Infrastructure Providers (VSIPs): VSIPs [8] deliver virtual service infrastructure to SPs, meeting particular service level requirements by combining physical network and cloud resources into service infrastructure meeting particular SLA requirements implemented through NFV-enabled network applications. These network applications might involve resources (or network functions) that are either implemented in traditional network hardware, or as virtualized NFs. These are the result of an orchestration system that interacts with the network control system as well as the cloud control system.

Infrastructure Providers (InPs): InPs own and maintain the physical infrastructure and run the virtualization environments. By virtualizing the infrastructure, they open up their resources to remote parties for deploying VNFs. The reusable physical resources comprise all possible resource options (computing, storage, and networking), and they span the entire service delivery chain from the end-user gateway and set-top-box over the access, aggregation, and core network up to the cloud.

Hardware Vendors: Hardware vendors provide the physical devices that are deployed by the infrastructure providers. The shift away from specialized equipment toward reusable, industry-standard high-volume servers, switches, and storage devices can reduce the total costs of infrastructure providers as they cost less than manufacturer-designed hardware and increase flexibility. The hardware must provide an interface toward the controller systems.

Software Vendors: Software vendors, including OSS developers, deliver the implementation of the logic that is used to optimally deploy the services on the physical infrastructure. Today a patchwork of specialized software products exists to realize that functionality. The most relevant software for the SDN/NFV architecture are those that focus on the following:
- The acceleration of packet processing on commodity hardware.
- Virtual machine technologies and software container-based technologies.
- Network virtualization software for virtualizing SDNs.
- SDN and cloud control software.
- Software for the orchestration of VNFs.
- Software implementations of VNFs.
- Software for monitoring, management, automated roll-out, configuration, and specification of VNFs.

For each of these, OSS communities have developed or are developing viable alternatives to proprietary software. We do not list all of these OSS projects due to space constraints (interested readers should consider [10] and the references therein).

Standards Development Organizations and Industry Fora: The networking industry today is very much standards-driven to make a product or service safe (safety standards) and interoperable (interface standards), while making the industry as a whole more efficient. The purpose of SDOs/IF such as ITU-T, ETSI, the Open Networking Foundation (ONF), IETF, the TM Forum, and the Metro Ethernet Forum (MEF) is to standardize the concepts that emerge in the ecosystem via coordination of the different actors in the development of new technical standards, as well as the revision and amending of existing standards when needed. Participants from across the ecosystem contribute to the development of these standards.

Next, we look into the details of the roles of

Software vendors, including OSS developers, deliver the implementation of the logic that is used to optimally deploy the services on the physical infrastructure. Today a patchwork of specialized software products exists to realize that functionality.

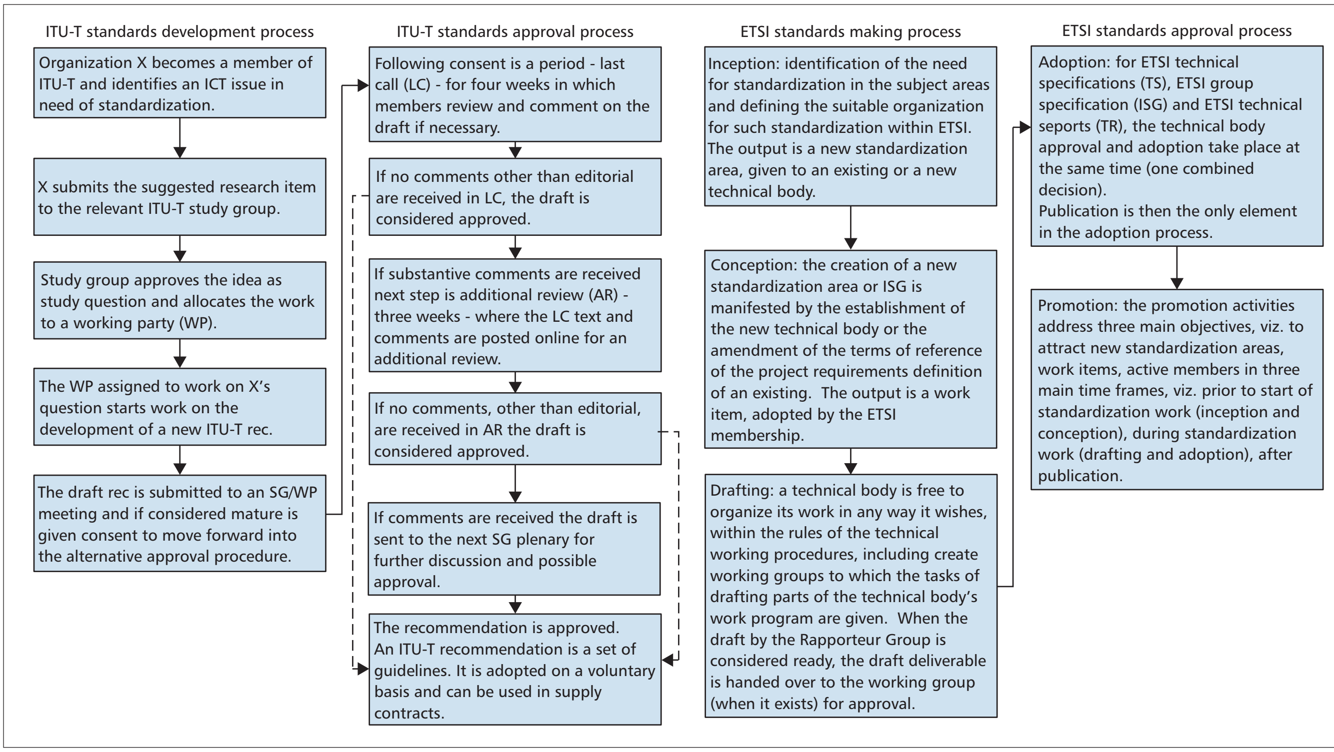

Figure 3. Overview of the standards development process and approval process of ITU-T and ETSI.

OSS communities on the one hand, and SDOs and industry fora on the other, in the development of standards.

Standards Developed by SDOs vs De-Facto Standards as a Result of the Work Done in OSS Communities

ETSI defines a standard as a document, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines, or characteristics for activities or their results, aimed at achievement of the optimum degree of order in a given context [9]. In general, five steps can be recognized in the standards process:

- Identification of the need.
- Assignment to the relevant body/group.
- Drafting and submission of the standard.
- Approval.
- Adoption and distribution.

The specific implementation differs between SDOs/IF, as illustrated in Fig. 3 for ITU-T and ETSI.

In practice, this requires significant time and effort due to:

- The difficulty of creating specifications of high technical quality.
- .The need to consider the interests of all of the affected parties.
- .The importance of establishing widespread community consensus.
- .The difficulty of evaluating the utility of a particular specification for the Internet community [11].

This is in sharp contrast with today's rapid development of networking technology, which demands the timely development of standards.

An OSS project, on the other hand, must deliver a working product. During the development, a de-facto market-based standard is created (development and standardization are executed as parallel processes). The agile development model, which is tied closely together with OSS projects, results in smaller incremental releases with each release, building on previous functionality. This approach takes into account that user demand is dynamic and that plans are short-lived. The OSS community decides on a way to implement a feature and, once it is included in the OSS project, it can be deployed at once. As a result, the opportunity exists to reduce the time-to-market. Similarly, SDOs/IF could apply an agile development approach in specification development to reduce their cycle time. For example, the authors in [12] state that the cycle time of a paper standard compared to an OSS project can be shortened by at least a factor of two.

SDOs focus on the design of norms or requirements of technical systems to achieve a technical goal that can only be met when multiple partners agree, and preferably subsequently adopt the proposed norm. Most SDOs follow a rigid specification mechanism, which once published, can only be corrected, changed, or extended in rather discrete steps following a rigorous process of validation and agreement. This makes SDO-based standards slow to adapt to a changing environment or problem statement. On the contrary, OSS projects are able to almost continuously adapt and integrate new code contributions driven by contributors in order to solve important current issues. While OSS communities can contribute to the goals of operators to reduce the costs of services and time-to-market, it should also be clear that the number of failed or dormant OSS projects is also notable [13]. Operators that want to contribute to OSS com-

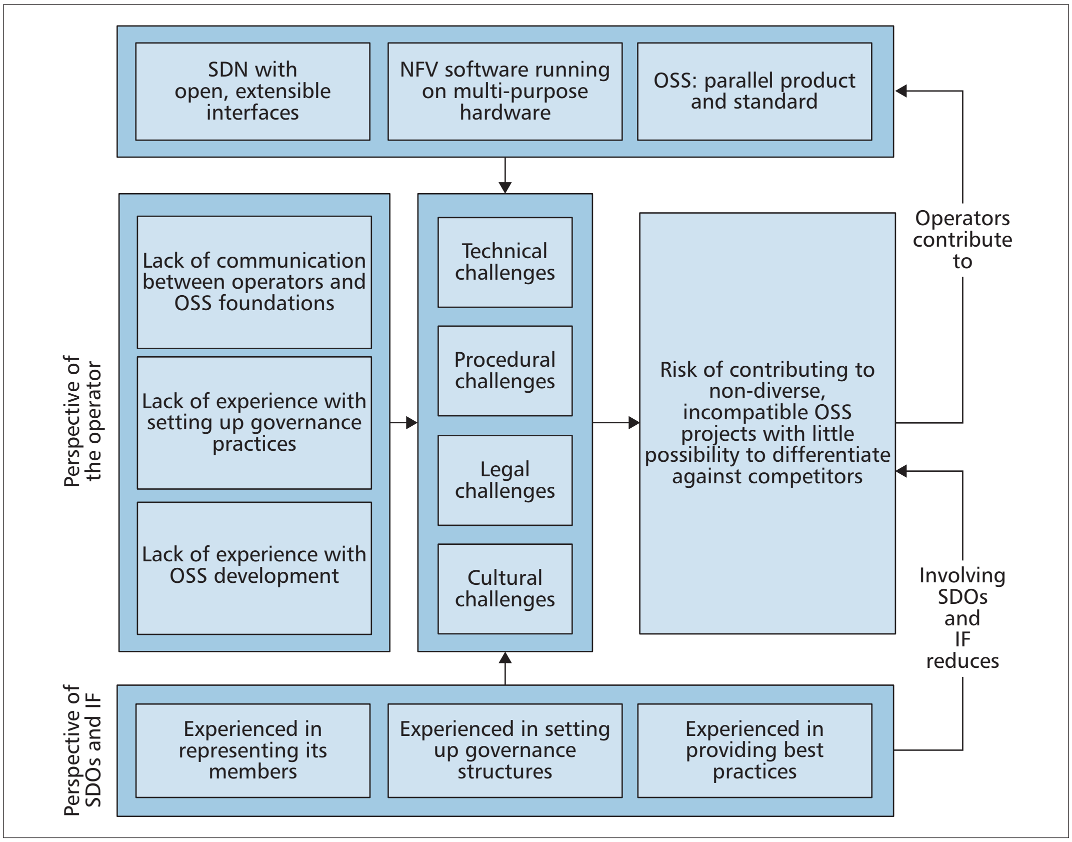

Figure 4. Interaction between operators, SDOs, IF and OSS foundations.

munities must therefore also overcome a variety of challenges [14]: technical, procedural, legal, and cultural.

Technical: OSS development can be disorganized as developers work on the parts that interest them most. Less tempting, but necessary, parts such as writing code documentation, automated tests, and manuals may as such receive less attention Also, to overcome fragmentation, OSS projects need to be able to interconnect and fit into a larger architecture. These technical challenges, while being pertinent to reach success, may receive less attention due to community diversity.

Procedural: From a procedural perspective, OSS cannot prevent companies from dominating a project and pushing though their own approach. This is a result of the lack of governance structure that ensures quality in the development and integration as well as the procedures for its assessment.

Legal: The choice of license may affect interoperability and the possibility for SPs and vendors to differentiate themselves. Permissive licenses, such as the Apache License Version 2, do not impose special conditions on the second redistribution, while strong licenses impose conditions in the event of wanting to redistribute the software. These conditions are intended to ensuring compliance with the license's conditions following the first distribution. Under the General Public License (GPL) of the GNU project, for example, it is only possible to redistribute code licensed under a compatible license, while under the Apache License Version 2, a project may be forced to develop proprietary extensions based on the material.

Cultural: As the center of value shifts from hardware toward software, the operator's culture and skillset must evolve as well (interested readers should consider [15] and the references therein). Operators typically work with product managers, while OSS communities focus on use cases and feature sets. Changing a company's culture is not a simple challenge, as internal resistance from people who fear losing their job can be severe when not properly managed.

To summarize this section, we wish to point at the conflicting goals of timely development of products and services on the one hand, and technical excellence, openness, and fairness on the other. Moving in one direction often leads to compromising in the other. Therefore, the next section focuses on how SDOs, IF, and OSS communities can work together to balance these conflicting goals and reach the common goal of creating flexible and efficient telecommunications networks.

Operators typically work with product managers, while OSS communities focus on use cases and feature sets. Changing a company's culture is not a simple challenge, as internal resistance from people who fear losing their job can be severe when not properly managed.

Guidelines for Improving Interaction Between OSS Communities, SDOs, and IF

Both SDOs and IF should engage with the OSS community to tackle the technical, procedural, legal, and cultural challenges that operators face in contributing to OSS. The causes of these challenges can be backtracked to a lack of communication, governance practices, and inexperience with OSS development.

The fundamental reason behind the existence of SDOs/IF is to avoid miscommunication and to establish impartial third-party governance practices. The competencies that SDOs/IF have

In parallel with the standards development process, code should be developed to support extensibility and modularity, and allow agile workflows for each of the modules independently. The NFV Proof of Concept-zone is an example of how function demonstration can be encouraged.

developed by performing these functions can provide an answer to the challenges that operators face when contributing to OSS. However, without change, the relevance of the interaction between SDOs, IF, and OSS will remain negligible. Attempts to bridge the gap via an alternative SDO model are therefore emerging. The ONF is an early example, which is dedicated to the promotion and adoption of SDN through open standards development. Initially established to promote the OpenFlow protocol via market development, the ONF now covers a broad range of specifications activities that encompass SDN architecture, the open common information model of network resources, a data model, and API development (including NETCONF, YANG, etc.). For example, to enable SDN control and network programmability, and allow SDN to be applied to a wide range of network resources, the ONF has a major effort to establish a consistent description of network resource functionality, capabilities, and flexibility. This resource description is provided by an information model that is independent of implementation details (including the protocol), providing the foundation with the derivation of a coherent suite of interface protocol-specific data models. Promoting a common industry-wide open model has been an informal collaboration among the ONF, ITU-T SG15, and the TM Forum. Between this, data model/API development, associated OS projects, and usage of OS tooling, ONF links these areas together in creating a bridge between "paper specifications" and "software development." Open Source SDN (OSSDN) is one example of how the ONF supports and sponsors OSS development by supplying people, monetary support for the maintenance and development of the community, and the hiring of a community manager. The Atrium project, which integrates OSS components and tries to make it easier for network operators to deploy SDN, is a direct outcome of that support. Another example is the Open Platform for NFV (OPNFV), a project operating under the Linux Foundation in close collaboration with ETSI's NFV ISG (among others), which has as its purpose the establishment of an integrated, open-source reference platform that uses the open-source NFV building blocks that already exist. A final example is the ETSI NFV Proof of Concept-zone, which promotes multi-vendor open ecosystems integrating components from different players.

To return to the goal of this article, we conclude the article by formulating a set of guidelines, based on lessons learned from alternative SDO models, which provide an outline toward what SDOs/IF can do to tackle the previously described challenges.

• SDOs/IF and OSS communities should establish open communication to reach more engagement in compatible projects. As an example, OpenMANO is an open source project (initiated by Telefónica) that provides a practical implementation of the reference architecture for management & orchestration under standardization at ETSI's NFV ISG (NFV MANO).

• SDOs/IF should emphasize software development and function demonstration more in its culture and structure by aligning their processes with the OSS development practices. In parallel with the standards development process, code should be developed to support extensibility and modularity, and allow agile workflows (e.g. hackathons) for each of the modules independently. The NFV Proof of Concept-zone is an example of how function demonstration can be encouraged.

• SDOs/IF should help OSS communities with the development of governance structures to guarantee technical excellence, openness, and fairness among the contributors to OSS projects. First, SDOs/IF should provide internal project governance in terms of developing the practices as well as the procedures that guarantee an effective development, integration, release, maintenance, and update process. and help in setting up the essential legal, business, management, and strategic processes. Second, SDOs/IF should offer cross-project governance to avoid:
- Unintentional competition between OSS projects that aim for the same goal (assuring project diversity).
- OSS projects that each deliver part of an overall solution, and which cannot be used together (assuring interoperability).

This is particularly challenging as these governance structures and processes differ among SDOs. In fact, it would also require an SDO/IF requirement upon an overall (modular) management/control architecture for software development in the domain of interest, with supporting guidelines, processes, and common open source tooling. This would assure consistency when diverse teams work independently on a part of the solution (e.g., technology-/application-/etc.-specification modules).

• SDOs/IF should guide operators, which are typically not so familiar with the world of OSS, among the plethora of OSS projects, and help them find the projects that best fit their needs and are worth contributing to. Examples are the Atrium and OPNFV projects, which integrate several OSS projects to speed up adoption.

• SDOs/IF should gather end users together, facilitate their discussions, and help operators with the definition of use cases and feature sets in a way that is implementable by an OSS project. As an example, OPNFV helps operators understand how to articulate their use cases as functional gaps in OSS projects.

• SDOs/IF should provide best practices in OSS development via training and learning materials, for example, by providing advice on best practices with regard to OSS licenses. SDOs/IF can help to make OSS credible for both operators and vendors (by preserving their ability to differentiate). For instance, OPNFV is licensed under an Apache 2.0 license, which explicitly grants patent rights where necessary to operate, modify, and distribute the software.

• SDOs/IF should overlook the integration of OSS projects and point toward development gaps while establishing and maintaining communication with other SDOs and IF. An example is the TM forum Catalyst proof of concepts, which bring together service providers and suppliers to work collaboratively. Another initiative, started by MEF and the TM Forum, is the UNITE program, to ensure a more open and rapid alignment of SDO work.

Summary

In this aticle we argued that margin pressure and the lack of possibilities for SPs to introduce new services has spurred their interest in:

- Emerging technologies such as SDN and NFV that provide an opportunity to reduce cost and increase flexibility.
- Other collaboration models such as OSS projects that can reduce the time-to-market.

By linking the most relevant ecosystem roles on the proposed overarching SDN/NFV architecture, we illustrated the general trend toward OS, particularly extensible APIs, in the SDN/NFV network space. Next, we focused on how these evolutions are changing the role of SDOs and IF, and how the OSS development methods affect how new standards are proposed, developed, and implemented. On one side of the spectrum, consensus-based standards developed by traditional SDOs tend to have a longer cycle time than the pace at which technology evolves. On the other side, OSS projects lead to a de-facto market-based consensus in a shorter cycle time. As such, SDOs may gradually lose their relevance in enabling innovation, and operators might turn to OSS communities to realize innovation. However, SPs that wish to contribute to OSS communities face technical, procedural, legal, and cultural challenges. We argue that the fundamental reason behind the existence of SDOs/IF is to resolve these challenges. Based on lessons learned from the interaction that is starting to happen between SDOs, IF, and OSS communities, we formulated a list of guidelines to improve interaction between both worlds and improve the relevance of SDOs/IF in innovation and increase the technical excellence, openness, and fairness of OSS projects.

Acknowledgements

We are grateful to the anonymous reviewers for their constructive comments and suggestions, which helped us to hone the core message of this article. This work has been partially funded by the European Commission under the 7th Framework research program project UNIFY.

References

[1] K. Pentikousis *et al.*, "Mobileflow: Toward Software-Defined Mobile Networks," *IEEE Commun. Mag.*, vol. 51, no. 7, July 2013, pp. 44–53.
[2] R. Bush, "Into the Future with the Internet Vendor Task Force: A Very Curmudgeonly View of Testing Spaghetti: A Wall's Point Of View," *ACM SIGCOMM Computer Commun. Rev.*, vol. 35, no. 5, Oct. 2015, pp. 67–68.
[3] C. Grundemann and J. Zorz, "Operators and the IETF," IETF Secretariat, Apr. 2015, available at https://datatracker.ietf.org/doc/draft-opsawg-operators-ietf/.
[4] ITU-T Recommendation Y2.621, "Requirements for an Independent, Scalable Control Plan in Future, Packet-based Networks," Aug. 2011, available at https://www.itu.int/rec/T-REC-Y.2621-201108-I/en.
[5] S. Sezer *et al.*, "Are We Ready for SDN? Implementation Challenges for Software-Defined Networks," *IEEE Commun. Mag.*, vol. 51, no. 7, July 2013, pp. 36–43.
[6] ONF, "TR-518 Relationship of SDN and NFV," Oct. 2015, available at: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/onf2015.310_Architectural_comparison.08-2.pdf.
[7] R. Szabo, "D2.2 Final Architecture," Nov. 2014, available at: https://www.fp7-unify.eu/files/fp7-unify-eu-docs/Results/Deliverables/UNIFY%20Deliverable%202.2%20Final%20Architecture.pdf.
[8] S. Latré *et al.*, "The Fluid Internet: Service Centric Management of Virtualized Future Internet," *IEEE Commun. Mag.*, vol. 52, no. 1, Jan. 2014, pp. 140–48.
[9] E. Soederstroem, "Standardising the Business Vocabulary of Standards" *Proc. 2002 ACM Symp. Applied Computing*, Mar. 2002, pp. 1048–52.
[10] W. Tavernier *et al.*, "Can Open-Source Projects (Re)-Shape the SDN/NFV-Driven Telecommunications Market?," *it-Information Technology*, vol. 57, no.5, Oct. 2015, pp. 267-276
[11] S. Bradner, "The Internet Standards Process – Revision 3" RFC 2026, IETF, Oct. 1996
[12] D. Ward, "Open Standards, Open Source, Open Loop," *IETF Journal*, Mar. 2015.
[13] S. Lee, "Measuring Open Source Software Success," *Omega*, vol. 37, no.2, June 2009, pp. 426–38.
[14] T. Kredil *et al.*, "Open Source: How to Speed NFV and SDN Deployment," *TM Forum Quick Insigths*, July 2015.
[15] C. Meirosu *et al.*, "DevOps for Software-Defined Telecom Infrastructures, draft-unify-nfvrg-devops-03.txt, IETF Secretariat, Work in Progress, Oct. 2015, available at: https://datatracker.ietf.org/doc/draft-unify-nfvrg-devops/.

Biographies

Bram Naudts (bram.naudts@intec.ugent.be) received his M.S. degree in applied economics in 2011 from Ghent University (Belgium), where he is currently pursuing a Ph.D. in techno-economics in the Department of Information Technology (INTEC). His research interests include techno-economic evaluation of communication network architectures and services, with a focus on software-defined networking and network function virtualization in research projects such as SPARC and UNIFY.

Wouter Tavernier (wouter.tavernier@intec.ugent.be) received a M.S. in computer science in 2002, and a Ph.D. degree in computer science engineering in 2012, both from Ghent University. He joined the IBCN group of Ghent University in 2006 to research future Internet topics. His research focus is on software-defined networking, network function virtualization, and service orchestration in the context of European research projects such as TIGER, ECODE, EULER, UNIFY, and SONATA.

Sofie Verbrugge (sofie.verbrugge@intec.ugent.be) received M.S. and Ph.D. degrees from Ghent University, Belgium. Since 2008 she has been working as a researcher affiliated with iMinds, a research institute to stimulate ICT innovation in Flanders, where she coordinates the techno-economic research within the IBCN group. In October 2014 she was appointed an associate professor at Ghent University. Her main research interests include cost modeling and telecom service and network deployment planning.

Didier Colle (didier.colle@intec.ugent.be) is a full professor at Ghent University. He received a Ph.D. degree in 2002 and a M.S. degree in electrotechnical engineering in 1997 from the same university. He is group leader in the iMinds Internet Technologies Department. He is co-responsible for the research cluster on network modelling, design and evaluation (NetMoDeL). This research cluster deals with fixed Internet architectures and optical networks, green-ICT, design of network algorithms, and techno-economic studies.

Mario Pickavet (mario.pickavet@intec.ugent.be) is a full professor at Ghent University, where he is teaching courses on discrete mathematics and network modeling. He is co-leading the research cluster on network modeling, design and evaluation (NetMoDeL). His main research interests are Internet architectures and optical networks, green ICT, and design of network algorithms. He is involved in several European and national research projects. He has published approximately 300 international publications in journals and in conference proceedings.

SDOs/IF should gather end users together, facilitate their discussions, and help operators with the definition of use cases and feature sets in a way that is implementable by an OSS project. As an example, OPNFV helps operators understand how to articulate their use cases as functional gaps in OSS projects.